INTRI: Contour-Based Trilateration for Indoor Fingerprint-Based Localization

Suining He and S.-H. Gary Chan, *Senior Member, IEEE*

Abstract—Due to its accuracy, trilateration has been widely deployed to locate smartphones outdoors. However, such approach cannot be easily applied indoors due to issues like non-line-of-sight measurement and complex multipath fading. Though fingerprinting overcomes these issues, its accuracy is often hampered by signal noise and the similarity metric comparing signal vectors. We propose INTRI, a novel, simple, accurate, and effective indoor localization framework combining strengths of trilateration and fingerprinting. Given a signal level received from an access point (AP) at target, INTRI first forms a contour given by reference points (RPs) with the same signal level, taking into account signal noise. The target is hence at the juncture of contours formed by all APs. We present selecting RPs for random signal by a width parameter determining the signal contour width (or spread). Then, an LP-based formulation finds the location following spirit of trilateration, which minimizes distance between target position and all contours. A novel particle filter leverages crowdsourced user inputs to adaptively estimate the width parameter. An online algorithm is further used to calibrate heterogeneous smartphones. Our extensive experiments in an airport, a shopping mall, and our campus show INTRI outperforms recent schemes with substantially lower error (often by more than 20 percent).

Index Terms—Indoor localization, contour-based trilateration, user-assistance, particle filter, fingerprinting, device calibration, signal contour, linear programming

1 Introduction

In trilateration, a mobile device (target) first computes its distances to a number of landmarks of known locations, often based on a signal-distance model. It then estimates its position which best matches these computed distances (by, for example, minimizing the error between the computed distances and the distances from that position to the landmarks). Such localization approach has achieved reasonable accuracy outdoors, with applications notably in Global Positioning System (GPS) and cellular positioning, where the landmarks are satellites and base stations (cell towers), respectively.

Despite of its successful outdoor application, trilateration does not work well indoors because distances to landmarks cannot be computed accurately and easily with models. Such inaccuracy is mainly due to non-line-of-sight landmarks, complex indoor signal environment (due to fading and multipath), over-simplification or parametric uncertainties in indoor propagation models, etc. Fingerprinting, on the other hand, emerges as a promising approach for indoor localization. An example is Wi-Fi fingerprinting, which is gaining popularity due to its ease of deployment without the need to install extra sensor infrastructure beyond the existing Wi-Fi network [1], [2].

Fingerprint-based localization is usually conducted in two phases. In the first offline (survey) phase, a site survey is conducted to collect the vectors of *received signal strength indicators* (RSSIs) at known locations, the so-called "reference points" (RPs). These vectors of RSSIs are the *fingerprints* of the locations and are stored in a database. In the second online (query) phase, a user samples or collects an RSSI vector at his own position and reports it to the server (in this paper, we use "user", "mobile device" and "target" interchangeably). The server then locates the indoor user by comparing the target vector with the fingerprints using some similarity metric such as euclidean distance [3] between signal vectors. The target position is then estimated out of the most similar "neighbors", the set of RPs whose fingerprints closely resemble the target's RSSI. (Note that the position may be estimated at the mobile device if it has enough resources.)

Traditionally, the similarity metric used in comparison often treats the RSSI vector of the target as a single "entity". This makes the neighbor selection susceptible to statistical signal fluctuation and measurement noise. Due to signal randomness, it has been widely observed that the matching in the online phase may be ambiguous, resulting in a dispersed set of neighbors (i.e., RPs of similar RSSI vectors are quite distant apart in the physical space). This leads to unsatisfactory localization accuracy [1], [4].

In this paper, we propose INTRI (**in**door **tri**lateration using signal contours), a novel, efficient and accurate indoor localization technique which employs the concept of trilateration in fingerprint-based environment. By treating the RSSI from each AP *individually* (instead of as a single signal vector), INTRI does not suffer from matching ambiguity. It is hence highly robust to random signals (due to signal fluctuation and measurement noise) as it is not based on fingerprint similarity comparison.

We illustrate the basic concept of INTRI, a contour-based trilateration approach, in Fig. 1 with three access points (APs). Let S be the signal strength of a certain AP as

• *The authors are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong. E-mail: {sheaa, gchan}@cse.ust.hk.*

Manuscript received 1 Mar. 2016; revised 18 July 2016; accepted 21 Aug. 2016. Date of publication 31 Aug. 2016; date of current version 3 May 2017. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2016.2604810

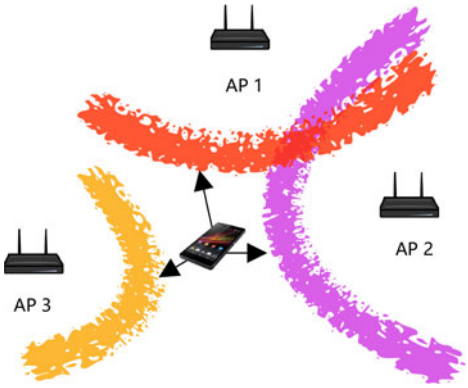


Fig. 1. The basic idea of indoor contour-based trilateration by minimizing the distance to signal contours.

measured by the target. For that AP, we can form a *contour* corresponding to S in the region, which is simply the set of spatially distributed RPs whose signal level for the AP is S, subject to its own signal randomness (the figure shows a continuous contour, while in reality the signal contour consists of discrete points in space). It is clear that the target is somewhere on the contour line. Given the target's received signal vector, we can hence form the contour for each of the APs (three contours for the three APs in the figure). Following similar spirit of trilateration, the target position can then be estimated by minimizing its distances to these contours.

INTRI integrates the trilateration technique with the indoor fingerprinting by combining the strengths of both approaches: it does not need the positions of APs and line-of-sight (LoS) measurement, and overcomes the dispersion problem by locating the target at the contour junction formed by its measured signal levels. Note that INTRI is a general approach applicable to any other fingerprint signal like Wi-Fi, Bluetooth Low Energy (BLE), channel state information (CSI), visible light, or RFID. We have further implemented INTRI over Wi-Fi and BLE to validate its generality.

In order for INTRI to be practically deployed, we need to overcome several critical challenges. First, how to select the set of RPs in a contour for random signals? This is by means of a width parameter to determine the width (or spread) of the signal in contour construction. Second, given the contours, how to localize efficiently and accurately the target? Third, note that signal randomness may vary over time and sites, and different devices may report different values on a certain signal (i.e., the heterogeneous device case). Beyond training contour parameter based on offline-collected fingerprint samples, we hence need to address how to adaptively and promptly adjust the width parameter over time in the site, and to "calibrate" the target device to match the signal level stored in the database.

In this paper, we make the following contributions in INTRI, addressing its accuracy and deployment issues:

- *Contour formation for random signals*: Signal measurement is inherently noisy. Constructing contours needs to consider such random fluctuation in order to accurately locate the target. Given fingerprints, we present how to statistically analyze them so as to construct contours under random signals.
- *Efficient contour-based localization algorithm*: We propose a novel contour-based localization algorithm based on linear programming formulation. Following similar spirit of trilateration, INTRI estimates target location with the objective to minimize the distances to the contours constructed according to the above.
- *Crowdsourced parameter estimation*: To make INTRI [5] more adaptive to signal characteristics over time and site, we propose a novel online algorithm based on *crowdsourced user input* and *particle filter* to better estimate the width parameter for contour construction. In our scheme, the crowdsourced user inputs can be easily or conveniently obtained through many approaches, such as explicit user-input locations [6], location-based social network, beacon assistance [7] or QR code scanning or NFC tags [8]. The contour parameter is first initialized as input to a particle filter. Then the particles which consistently match the crowdsourced location inputs are selected to generate appropriate signal width in the contours for localization. This achieves less over-fitting upon the offline fingerprint samples, and better robustness under noisy environment and over time.
- *Adaptive device calibration*: Regarding device heterogeneity issue (for the case that the devices in the online and offline phases may be different), offline calibration for all devices is neither practical nor scalable. We propose an *adaptive online* algorithm to efficiently calibrate devices, which adjusts the target measured RSSI according to the stored fingerprints based on signal correlation. Using such an approach, INTRI achieves scalability for all current and emerging heterogeneous devices.

We have conducted extensive simulation and large-scale experimental trials in the Hong Kong International Airport (HKIA), the Hong Kong Cyberport mall (HKCP) and our university campus (HKUST). Both simulation and experimental studies further confirm the high accuracy and environmental adaptivity of INTRI.

The rest of this paper is organized as follows. After reviewing the related works in Section 2, we describe in Section 3 the system framework and forming contours for noisy signals. Based on these contours, in Section 4 we present the linear programming formulation for localization based on the idea of trilateration. Then in Section 5, we discuss the idea of improving contour settings with crowdsourced inputs, followed by an efficient online device calibration algorithm. Illustrative simulation and experimental results are presented in Section 6. We finally conclude in Section 7.

2 Related Work

We briefly discuss related works in this section. Pattern recognition techniques have been widely studied in Wi-Fi fingerprinting localization. RADAR [3] and Horus [9] are the two representative approaches. Recently more advanced techniques on pattern matching have been investigated [1], [10], [11], [12]. Signal propagation model is also considered recently to derive RSSI at different locations [2], [13]. EZ [13] utilizes rigid matching between signals and distances to determine the target location. More recent works like

TABLE 1
Major Symbols Used in INTRI Formulation

Notation	Definition
N, L	Number of RPs & APs in the fingerprint database.
$\mathbf{f}_n$	Wi-Fi RSSI vector received at RP n.
f_n^l	Wi-Fi AP l's RSSI at RP n (dBm).
$\mathbf{g}$	Wi-Fi RSSI vector received at target.
g^l	Wi-Fi AP l's RSSI at target (dBm).
σ_n^l & σ^l	AP l's standard deviation at RP n & the target (dB).
z_0	Width parameter for contour construction.
$\mathbf{S}^l$ & $\mathbf{C}^l$	Signal contour from AP l & the indices of RPs on it.
$\widehat{d}_l$	Pseudo distance from AP l.
D_n	Contour weight at RP n.
$\mathbf{R}$	Indices of selected RPs for LP localization.
$N^{\mathbf{R}}$	Number of RPs selected in $\mathbf{R}$.
Δ_n^l	Minimum distance between RP n and $\mathbf{S}^l$.
ω_n	Weight of RP n in target estimation.
$\mathbf{r}_n$	2-D coordinate of RP n.
$\widehat{\mathbf{x}}$	Estimated 2-D coordinate of target.
$\mathbf{x}^i$	Crowdsourced input location.
ψ^p	Distance between user-input location and particle p
θ_p	Weight of particle p in user assistance.

EZPerfect [14] and Modellet [2] further utilize the labeled fingerprints to derive the signal propagation model.

In contrast to the above, we combine the strengths of fingerprinting and trilateration approaches. We employ a geometric scheme (following the similar spirit of trilateration) based on random fingerprint signals to constrain the target region. By formulating a novel linear programming, INTRI achieves much better localization accuracy without neighbor dispersion.

Some other recent works leverage the temporal or spatial RSSI patterns for localization. These works consider location-dependent patterns such as the trend of RSSI sequences along corridors [15], order of RSSIs from different APs [16], or the unique existence of some Wi-Fi APs at some area [17]. Once the target measures such patterns, its location is then mapped to the area. These patterns achieve promising results for constrained and narrow environment with well-defined user trajectories (like corridors or offices). In contrast, the contours in INTRI are solely derived from fingerprints and are applicable to any indoor environment. Furthermore, INTRI does not need the positions of APs and LoS measurement, and is not based on indoor models (indoor environment may be too complex to model).

Calibrating different devices has been studied in recent works [18], [19], [20], [21]. Traditional offline calibration [18] causes extra manual efforts in real deployment and hence is not scalable. Given the target RSSI measurement, works like [19], [20], [21] utilize the deduction [19] (or ratio [20]) between AP signal values to calibrate the devices. However, large noise and fluctuation in signal levels can degrade the quality of above calibration approaches. Some learning-based approaches [2] utilize expectation maximization to calibrate the signal difference. Different from these works, INTRI proposes a more efficient and robust scheme which maps the target signals to the signal space in fingerprint database.

User-input has been recently studied in [22], [23] for better indoor localization. Many previous works only focus on improving the fingerprint database [6], [24] or calibrating heterogeneous devices [22]. Unlike these works, to our best knowledge, INTRI is the first work calibrating location-based system parameters with user assistance. The user simply inputs his/her current location if a location correction is needed for INTRI. By implementing a novel particle filter algorithm [25], INTRI learns the suitable contour width parameters for the environment. Hence our INTRI becomes adaptive to practical deployment on new sites. Our study of parameter estimation is orthogonal and also amendable to crowdsourcing feedback works in [26], [27], [28], and can serve as a plug-in for many other localization systems [3], [9] for better adaptivity.

To improve fingerprint localization, sensor fusion has attracted intensive attention recently. Using the smartphone inertial sensors, fusing motion information has been studied extensively to improve fingerprint-based positioning [1], [29]. Different from above fusion approaches, INTRI solely relies on the wireless measurement (such as Wi-Fi and iBeacon), and therefore does not require inconvenient motion or step length calibration. INTRI is also independent of these works, and is amendable to integrate with them for higher accuracy of mobile localization.

A preliminary version of INTRI has been reported in [5]. In this paper, we further advance from it in the following major ways: 1) We present a novel approach based on particle filter and crowdsourced input (say, locations and signals from user feedbacks or sensors) to adaptively and promptly adjust the width parameter in contour construction. This improves localization accuracy. We have conducted extensive experiments in different sites to validate the improvement and robustness of such crowdsourced approach. 2) Beyond Wi-Fi, we have deployed INTRI with iBeacon, and conducted extensive experiments to show the more general applicability of INTRI. We also compare INTRI with recent state-of-the-art algorithms to validate its accuracy improvement for iBeacon-based localization.

3 System Framework & Forming Signal Contours

In this section, we describe how to properly form the signal contours for later location estimation. After presenting system framework in Section 3.1, we present in Section 3.2 how to find the signal contour for each AP, given target random RSSIs. A contour (of an AP) consists of discrete RPs whose signal level is the same as the target's measured signal level, subject to statistical fluctuation. To achieve higher localization accuracy, the RPs visited by many contours of strong signals are preferred. Therefore, in Section 3.3 we propose a weighting scheme which is able to differentiate the importance of the RPs. The important RPs are kept while those unimportant ones (where target is unlikely at) are filtered. After the above steps, given that the target is likely to be in the "dense" region of selected RPs, we finally present in Section 3.4 how to find such region based on maximally connected components. The major symbols used in this paper are shown in Table 1.

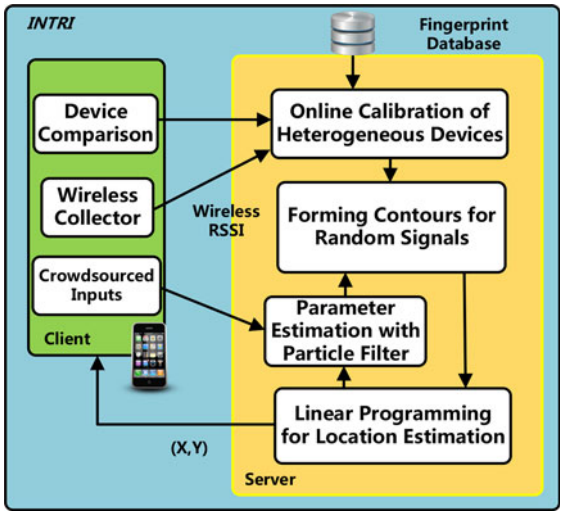


Fig. 2. System framework of INTRI.

3.1 System Framework

We show in Fig. 2 the system framework of INTRI. The Wi-Fi fingerprint database is initialized by a site survey, storing $<$location, RSSI vector$>$ pairs for each RP and vendor information of the devices used for data collection. The system is now ready for online estimation.

In the online phase, INTRI first checks the vendor information of the user's device. If that is different from the devices used for site survey, the target RSSI vector will be calibrated using the stored fingerprints. The calibrated RSSI vector is then used to construct the signal contours. Given signal contours, INTRI formulates a linear programming (LP) to jointly minimize the distances to the contours and estimates the user's position. The location is finally returned to the user's device. During the initial deployment of INTRI, crowdsourced inputs are fed to the particle filter learning module in order to learn the contour width parameter in the site. After the contour width parameter is adapted and learned, it is stored for online localization purpose.

3.2 Forming Signal Contour

Let N and L be the total number of RPs and distinct APs detected in the whole survey site, respectively. Further let $\widetilde{F}_n^l$ be the random variable of the RSSI collected at RP n for AP l in the offline fingerprint collection, where $1 \le n \le N$ and $1 \le l \le L$. Multiple samples are collected at different time τ indexed by $1, 2, \ldots, T_n^l$ for RP n and AP l, which are denoted as $\{f_n^l(\tau) | \tau = 1, \ldots, T_n^l, T_n^l > 1\}$, where T_n^l is the total number of samples collected.

The unbiased estimate of expectation $E(\widetilde{F}_n^l)$, denoted as $\widehat{\mu}_n^l$, is the mean of $f_n^l(\tau)$'s. The unbiased estimate on the variance of $\widetilde{F}_n^l$ is denoted as $\widehat{\sigma}^2(\widetilde{F}_n^l)$. Then $\widehat{\mu}_n^l$ and $\widehat{\sigma}^2(\widetilde{F}_n^l)$ are respectively given by

$$\begin{aligned} \widehat{\mu}_n^l &= \frac{1}{T_n^l}\left(\sum_{\tau=1}^{T_n^l} f_n^l(\tau)\right), \\ \widehat{\sigma}^2\left(\widetilde{F}_n^l\right) &= \frac{1}{T_n^l - 1}\left(\sum_{\tau=1}^{T_n^l}\left(f_n^l(\tau) - \widehat{\mu}_n^l\right)^2\right). \end{aligned} \tag{1}$$

Let f_n^l be the mean of the measured fingerprint signals (a random variable) at RP n for AP l, given by

$$f_n^l = \frac{1}{T_n^l}\sum_{\tau=1}^{T_n^l} F_n^l(\tau), \tag{2}$$

where $F_n^l(\tau)$'s are random variables distributed as $\widetilde{F}_n^l$ and $f_n^l(\tau)$'s are their realized values. Let $v(\tau)$ be a noise process independent from $f_n^l(\tau)$. Let α_n^l be a parameter determining the autocorrelation of samples. Then the signal time series can be represented as a first order autoregressive model [30],

$$f_n^l(\tau) = \alpha_n^l f_n^l(\tau - 1) + (1 - \alpha_n^l)v(\tau), \tag{3}$$

where α_n^l represents the correlation between successive samples ($0 \le \alpha_n^l \le 1$). For f_n^l, its expected value $\bar{f}_n^l$ and standard deviation σ_n^l [30] can be estimated as

$$\begin{aligned} \bar{f}_n^l = \widehat{\mu}_n^l, \quad \sigma_n^l = \Bigg\{ \frac{\widehat{\sigma}^2\left(\widetilde{F}_n^l\right)}{\left(T_n^l\right)^2} \Bigg[\left(\frac{1 - (\alpha_n^l)^{T_n^l}}{1 - \alpha_n^l}\right)^2 + T_n^l - 1 \\ - (\alpha_n^l)^2 \frac{1 - (\alpha_n^l)^{2(T_n^l - 1)}}{1 - (\alpha_n^l)^2} \Bigg] \Bigg\}^{1/2}, \end{aligned} \tag{4}$$

respectively. Here α_n^l can be approximated by autocorrelation coefficient with lag one [30] for RSSI samples in the offline fingerprinting, i.e.,

$$\alpha_n^l \approx \frac{\sum_{\tau=1}^{T_n^l - 1}\left(f_n^l(\tau) - \widehat{\mu}_n^l\right)\left(f_n^l(\tau + 1) - \widehat{\mu}_n^l\right)}{\sum_{\tau=1}^{T_n^l}\left(f_n^l(\tau) - \widehat{\mu}_n^l\right)^2}. \tag{5}$$

Such autocorrelation within sequential RSSIs may be related to the caching in Wi-Fi sampling. Wi-Fi data caching can be identified using Timing Synchronization Function (TSF) of the RSSI. We can conclude that an RSSI is a cached one if its TSF is identical to another in an earlier scan result.

Therefore, here we implement Equations (4) and (5) to calculate σ_n^l for more general cases. Given the above, let

$$\mathbf{f}_n = [\bar{f}_n^1, \bar{f}_n^2, \ldots, \bar{f}_n^L], \quad \boldsymbol{\sigma}_n = [\sigma_n^1, \sigma_n^2, \ldots, \sigma_n^L], \tag{6}$$

be the RSSI vector (fingerprint) and variance at RP n. This ends the offline phase in forming fingerprints for INTRI.

In the online query stage, denote the target measurement of AP l as g^l. We denote the RSSI vector at the target as

$$\mathbf{g} = [g^1, g^2, \ldots, g^L]. \tag{7}$$

For Equations (6) and (7), by definition, $\bar{f}_n^l = 0$ ($g^l = 0$), if AP l is not sampled at RP n (at the target).

As online g^l is also random signal, we utilize the uncertainty in offline fingerprint to characterize its variation. Specifically, let $(\sigma^l)^2$ be its variance, estimated as the mean of the variance in all the fingerprints, i.e.,

$$(\sigma^l)^2 = \frac{1}{|\mathbf{N}^l|}\left(\sum_{n \in \mathbf{N}^l} \widehat{\sigma}^2\left(\widetilde{F}_n^l\right)\right), \tag{8}$$

where $\mathbf{N}^l$ is the set of RPs detecting AP l in the site and $|\mathbf{N}^l|$ is its cardinality.

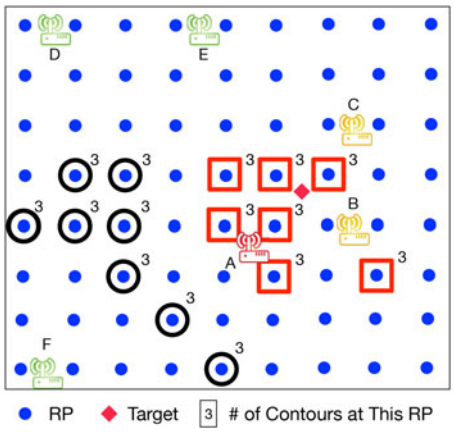


Fig. 3. Illustration of differentiating signal contours.

We consider the randomness in the difference between fingerprint and target RSSI, i.e., $g^l - f_n^l$, for each AP l. As g^l and f_n^l are independently measured, the variance of $g^l - f_n^l$ is therefore given by

$$\mathbb{V}(g^l - f_n^l) = \mathbb{V}(g^l) + \mathbb{V}(f_n^l) = (\sigma^l)^2 + (\sigma_n^l)^2. \tag{9}$$

By evaluating the signal map of each given AP within the survey site, we may observe a set of RPs which share similar RSSI values with target one, subject to some statistical fluctuation. These RPs forms the signal contour for that AP. Specifically, a *signal contour* for AP l, denoted as $\mathbf{S}^l$, consists of a set of RPs where the target is likely within. In other words, $\mathbf{S}^l$ represents the RPs whose RSSI for AP l is likely within a certain *range* from g^l. Therefore, in finding the contour $\mathbf{S}^l$, we *eliminate* the RPs whose RSSI is more likely far away from the target's, i.e., if

$$|g^l - \bar{f}_n^l| > z_0\sqrt{\mathbb{V}(g^l - f_n^l)}, \quad z_0 > 0, \tag{10}$$

where z_0 is termed *width parameter*, because $z_0\sqrt{\mathbb{V}(g^l - f_n^l)}$ together represents the width (or spread) of the signal from AP l at RP n to be included into the contour. z_0 determines the sensitivity of INTRI towards the signal noise. In deployment of INTRI, we can determine z_0 through offline training samples. We may initially assign a value for z_0 (say, 2 or 3 in our experiment). In order to make INTRI more adaptive to online signal environment, we further consider a novel algorithm learning z_0 online (Section 5.1).

Then a target measuring a similar RSSI value is likely within a certain range from that contour, i.e.,

$$g^l - z_0\sqrt{\mathbb{V}(g^l - f_n^l)} \le \bar{f}_n^l \le g^l + z_0\sqrt{\mathbb{V}(g^l - f_n^l)}. \tag{11}$$

Given AP l, we denote the corresponding index set of RPs on contour $\mathbf{S}^l$ as $\mathbf{C}^l$, i.e.,

$$\mathbf{C}^l = \{n \in \mathbf{N}^l, \text{where } \bar{f}_n^l \text{ satisfies Equation (11)}\}. \tag{12}$$

Then we denote its cardinality as $|\mathbf{C}^l|$.

3.3 Calculation of Contour Weights

Given the found contours, an intuitive idea is to locate the target at RPs with the maximum number of contours. However, due to the indoor partitions and signal measurement uncertainty, spatially dispersed RPs may have very similar number of signal contours passing by. In this case, finding the RPs with the largest number of contours may not lead to accurate location estimation. As shown in Fig. 3, a target (red diamond) measures a signal vector consisting of RSSIs from six APs, A, B, C, D, E, and F. The contours of D, E, and F (black circles) shift from those of A, B, and C (red rectangles) due to signal fluctuation or wall partitions. Thus, the RPs in black circles may share the same number of contours as those close to target. If all these RPs are considered equally without sufficient filtering or differentiation, large location errors still exist.

Through empirical studies, we observe that the strong signals near APs provide more reliable location-dependent information than the weak ones. The stronger the RSSI, the more important the AP contour is in contributing to location estimation. This is due to the sharpness of signal strength change at the locations near the Wi-Fi APs, which differentiates RPs the most from other distant ones. Inspired by this, we consider as follows how to utilize such distinguishable RSSIs to improve location accuracy.

Take Fig. 3 again as an illustration. The signals at contours of A, B and C are stronger since they are closer to the corresponding APs. If we can assign more weights on the contours with strong signals in final location decision, we can distinguish the important RPs more accurately. Thus, we propose a weighting scheme which differentiates RPs the most and finds the RPs with higher confidence. The physical intuition of the weighting scheme is based on the log-distance path loss (LDPL) model [2], [13]. We adopt it in the weighting function for signal contour differentiation, which achieves high localization accuracy.

Denote the reference power at distance d_0 as P_0^l (dBm). Let d_l be the distance between target and AP l. Then the received power at target from AP l is given by

$$g^l = P_0^l - 10\gamma^l \log_{10}\left(\frac{d_l}{d_0}\right) + X, \tag{13}$$

where γ^l denotes the decay rate of RSSI in propagation. X represents the inherent signal fluctuation and noise. Here we consider using LDPL to represent the closeness of APs for contour differentiation (not exact distance), and X has been considered separately in forming contours (Section 3.2). Based on Equation (13), we define the corresponding *pseudo distance* from AP l ($d_0 = 1$ m) as

$$\widehat{d}_l = 10^{\frac{P_0^l - g^l}{10\gamma^l}}. \tag{14}$$

Instead of indicating actual distances, we use it to represent the *confidence level* with AP signals. Specifically, the smaller $\widehat{d}_l$, the more likely the AP is nearby. Then the *contour weight* at RP n from all detected APs is defined as

$$D_n = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{\widehat{d}_l}. \tag{15}$$

Using Equation (15), for each g^l at contour $\mathbf{S}^l$, we consider the potential that an AP is physically nearby. The larger D_n, the more contours of strong signals hit the RP. Such APs are more likely to be close to these RPs around the target, and

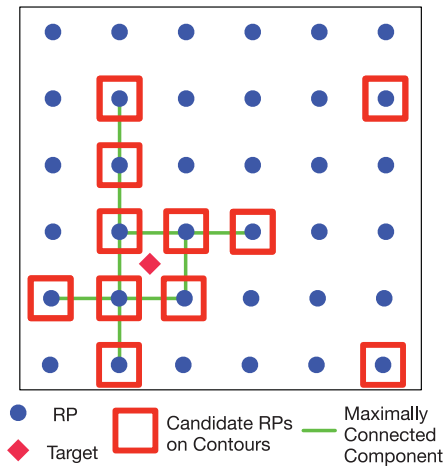


Fig. 4. Illustration of MCC to find the dense contour region.

we further utilize such closeness information to constrain the target region.

Then we find the RP set $\mathbf{R}$ consisting of the indices of RPs which have the highest contour weights as the potential area for final estimation, i.e.,

$$\mathbf{R} = \arg\max_{n} D_n. \tag{16}$$

In INTRI, we dynamically select the RPs with D_n higher than $\rho \max\{D_n\}$ ($\rho = 0.75$ in our simulation and experiment). RSSIs from an AP that is located in a region surrounded by obstacles may lead to larger γ^l than those from other APs with freer signal propagation. Here we do not assume ideal line-of-sight measurement. The external parameters (P_0^l and γ^l) in Equations (13) and (14) can be learned through gradient decent analysis over the fingerprint signals [13], [31].

To summarize, by traversing the survey site, we find the signal contours at each RP within the signal range in Equation (11), and calculate the contour weights at that RP using Equations (14) and (15). The most important RPs with strong signals will be selected to form $\mathbf{R}$.

3.4 Finding the Dense Contour Region

The selected RPs in $\mathbf{R}$ may still have "strayed RPs" due to measurement uncertainty. In Fig. 4, we illustrate the spatial distribution of $\mathbf{R}$ (red rectangles), which is based on the extensive experimental observations in the Hong Kong International Airport (HKIA) trials (from $1,100$ target RSSI samples). We can find RPs which are physically close to each other. These RPs form a region where the target is likely at, and should be used in our localization formulation. Due to signal temporal fluctuation, some RPs (the two to the right) exist and are distant away from the target location. If included in localization, these RPs may lead to location error and unnecessary computation. Considering their spatial connectivity, we observe in Fig. 4 that the RPs near the target (red diamond) form a connected component with the largest cardinality, namely the *maximal connected components* (MCC). Based on such observation, we utilize an algorithm of finding MCC, and find the region with dense contours by filtering out RPs not in the region.

We show the process of such filtering in Algorithm 1. Let $N^{\mathbf{R}}$ be the cardinality of $\mathbf{R}$. We first construct an $N^{\mathbf{R}} \times N^{\mathbf{R}}$ adjacency matrix $\mathbf{A}$, where $\mathbf{A}(i,j) = 1$ indicates that RP i and j are adjacent, and $\mathbf{A}(i,j) = 0$ otherwise (Lines 1). We set the threshold of adjacency as $\sqrt{2}$ times of the square grid width in site survey. By treating RPs in $\mathbf{R}$ as an undirected graph, we find the membership list of all connected components [32] (Lines 11 to 27). After that, we find the component with the maximum number of RPs (Line 28). If multiple components have the maximum cardinality in common, we use their union for later localization.

Algorithm 1. Finding Dense Contour Region

Input: $\mathbf{R}$: indices of selected RPs; ζ: threshold.
Output: $\mathbf{R}$: set of RPs with dense contours.
/* Constructing Adjacency Matrix. */
1: $\mathbf{A} \leftarrow zeros(N^{\mathbf{R}}, N^{\mathbf{R}})$. /* $N^{\mathbf{R}} \times N^{\mathbf{R}}$ matrix. */
2: **for** $i \in \{2, \ldots, N^{\mathbf{R}}\}$ **do**
3: **for** $j \in \{1, \ldots, i\}$ **do**
4: **if** $dist(i,j) \leq \zeta$ **then**
5: $\mathbf{A}(i,j) \leftarrow 1$; $\mathbf{A}(j,i) \leftarrow 1$;
6: **end**
7: **end**
8: **end**
9: $isDiscovered \leftarrow zeros(N^{\mathbf{R}})$;
10: $\mathbf{m} \leftarrow \{\}$; /* Set of Components */
11: $nG \leftarrow 0$; /* Number of Components */
12: **for** $n \in \{1, \ldots, N^{\mathbf{R}}\}$ **do**
13: **if** $!isDiscovered(n)$ **then**
14: $nG \leftarrow nG + 1$; $isDiscovered(n) \leftarrow 1$;
15: $\mathbf{m}[nG].pt \leftarrow \mathbf{m}[nG].pt \cup \{n\}$; $ptr \leftarrow 1$;
16: **while** $ptr \leq sizeof(\mathbf{m}[nG].pt)$ **do**
/* Find Its Neighbors. */
17: $nei \leftarrow find(A(:, \mathbf{m}[nG].pt(ptr)))$;
18: **for** $nb \in \{1, \ldots, sizeof(nei)\}$ **do**
/* Connected Components */
19: **if** $!isDisovered(nei(nb))$ **then**
20: $isDisovered(nei(nb)) \leftarrow 1$;
21: $\mathbf{m}[nG].pt \leftarrow \mathbf{m}[nG].pt \cup \{nei(nb)\}$;
22: **end**
23: **end**
24: $ptr \leftarrow ptr + 1$;
25: **end**
26: **end**
27: **end**
28: $\mathbf{R} \leftarrow MaxMembers(\mathbf{m})$;

In Fig. 5, we plot the histograms of $N^{\mathbf{R}}$ from $1,100$ targets in HKIA before and after the proposed RP filtering. We can observe that such a scheme narrows the search scope and facilitates the final location estimation.

4 LP for Location Estimation

In this section, we present the core formulation of INTRI. To formulate the objective function, we first define in Section 4.1 the physical (geographical) distance from an RP to signal contours. Then in Section 4.2, we formulate a linear programming based on weighted physical distances to those contours. We finally analyze the online computational complexity of INTRI in Section 4.3.

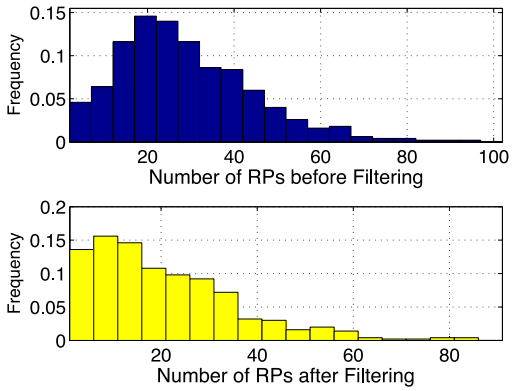


Fig. 5. Histograms of $N^{\mathbf{R}}$ before and after RP filtering (the airport data).

4.1 Defining Distances from an RP to Signal Contours

We are given a set of RPs where the contours locate. In the following we introduce how to utilize the signal contours as the objective for INTRI.

Recall that traditional trilateration estimates target position by minimizing the difference between the measured distances and the distances from the position to the landmarks. In our formulation, based on the same spirit, we use the distances to the constructed signal contours. As RPs on contours are discretely sampled in the survey site and a target is surrounded by RPs, we utilize in our formulation the distances from RPs in $\mathbf{R}$ to those on other contours. The RPs with small distances to other $\mathbf{S}^l$'s are likely to be the target region.

Let $\mathbf{r}_n = [x_n, y_n]$ be the coordinate of RP $n \in \mathbf{R}$. We calculate its distance from each RP m ($n \neq m$) in $\mathbf{S}^l$, i.e.,

$$\delta^l_{nm} = \sqrt{\left(x_n - x^l_m\right)^2 + \left(y_n - y^l_m\right)^2}, \quad \forall m \in \mathbf{C}_l, \tag{17}$$

where $[x^l_m, y^l_m]$ is RP m's coordinate on signal contour l. Given all distances $\delta^l_{nm}, \forall m \in \mathbf{C}_l$, we find the minimum one, i.e.,

$$\Delta^l_n(\mathbf{g}, \mathbf{f}_n) = \min \delta^l_{nm}, \quad \forall m \in \mathbf{C}_l, \tag{18}$$

which represents the distance between RP n and contour l. Given above, the distances to all contours are aggregated as

$$\Delta_n = \sum_{l=1}^{L} \Delta^l_n(\mathbf{g}, \mathbf{f}_n), \tag{19}$$

which, in other words, approximates the residual between the target estimation's distances to the landmarks and the measured distances.

By minimizing Equation (19), we minimize the distance difference and hence extend the idea of trilateration into a contour-based scheme. As the contours are derived from fingerprints and target RSSIs, we require no explicit knowledge of AP locations or LoS measurement, and therefore combine the advantages of both fingerprinting and trilateration in our formulation. In the following, we present the formulation to find the target position.

4.2 Linear Programming Formulation

Given $\mathbf{R}$ and Δ_n, we formulate the localization problem based on linear programming (LP). For each target, denote its estimated location as $\widehat{\mathbf{x}} = [\widehat{x}, \widehat{y}]$. Let ω_n be the weight assigned to $\mathbf{r}_n$ in locating the target. As the target is surrounded by the RPs in $\mathbf{R}$, its estimated position can be expressed as

$$\widehat{\mathbf{x}} = \sum_{n=1}^{N^{\mathbf{R}}} \omega_n \mathbf{r}_n, \tag{20}$$

where $\mathbf{r}_n \in \mathbf{R}$, and the weights satisfy the normalization and non-negativity, i.e.,

$$\sum_{n=1}^{N^{\mathbf{R}}} \omega_n = 1, \quad \omega_n \geq 0, \quad \forall n \in \{1, \ldots, N^{\mathbf{R}}\}. \tag{21}$$

Based on Equation (18) and (20), we extend the idea of trilateration into finding the weights which minimize the target's weighted sum of distances to all the contours, i.e.,

$$\arg\min_{\{\omega_n\}} \sum_{n=1}^{N^{\mathbf{R}}} \omega_n \Delta_n. \tag{22}$$

In real deployment, a target is far more likely to be between two neighboring RPs (or within the square grid formed by four RPs). In order to jointly consider the neighboring RPs to the target, we set a constraint over the weight ω_n at each RP, i.e.,

$$w_n \leq W, \quad \forall n \in \{1, \ldots, N^{\mathbf{R}}\}, \tag{23}$$

where W is a dynamic parameter determined by the maximum contour weight

$$W = \frac{\max D_n}{\sum_{n=1}^{N^{\mathbf{R}}} D_n}. \tag{24}$$

Through Equations (22) and (23), we can jointly consider the physical distances (denoted as $\{\Delta_n\}$) and the contour weights (denoted as $\{D_n\}$) in our formulation.

If there are indoor wall partitions in narrow space, we can include map constraints in our basic formulation. Denote the set of map constraints as $\mathbf{E}$. For each edge $e \in \mathbf{E}$, we consider the accessible area within the map constraints as

$$a_e \widehat{x} + b_e \widehat{y} + c_e \geq 0, \quad e \in \mathbf{E}, \tag{25}$$

where a_e, b_e and c_e are the line parameters obtained from the site map in our system initialization. The formula of map edges can be easily found using the nearest map constraints. Using the above, the localization problem can therefore be formulated as a linear programming (LP):

$$\begin{aligned} &\text{Objective: Equation } (22), \\ &\text{subject to: Constraints } (20), (21), (23), (24), \textit{and } (25). \end{aligned} \tag{26}$$

In other words, we opt to find the estimated position with the smallest weighted physical distances to the contours within the accessible area. INTRI returns the set of weights assigned to RPs in $\mathbf{R}$ which minimizes the distances to contours, i.e., the RPs which are closer to all contours are assigned higher weights, and vice versa.

Using some commercial optimization solver (say, JOptimizer [33] for Android implementation or CVXOpt [34] for server), we can solve the above LP efficiently [35]. The final solution $\{\omega_n\}$ is then used to estimate the target position with Equation (20). We summarize INTRI in Algorithm 2.

4.3 Complexity Analysis

We here analyze the computational complexity of INTRI:

1) *Forming contours with random signals:* Given N RPs and L APs, the complexity of finding signal contours and calculating contour weights is given by $\mathcal{O}(NL)$ (Sections 3.2 and 3.3).
2) *Finding maximally connected components:* Given $N^{\mathbf{R}}$ selected RPs, finding the maximally connected component takes $\mathcal{O}((N^{\mathbf{R}})^2)$ [32] (Section 3.4).
3) *LP-based localization:* To prepare the objective function of LP, calculating the distances between $\mathbf{R}$ and contours takes $\mathcal{O}\big(N^{\mathbf{R}}L|\mathbf{C}^l|\big)$ (Section 4.1). In Formulation (26), there are $\mathcal{O}(N^{\mathbf{R}})$ decision variables in $\{\omega_n\}$. Thus, the LP in location estimation takes weak polynomial time, i.e., $\mathcal{O}\Big((N^{\mathbf{R}})^3\Big)$ [35] (Section 4.2).

To summarize, the overall online running time of INTRI is

$$\mathcal{O}\Big(NL + N^{\mathbf{R}}L|\mathbf{C}^l| + \big(N^{\mathbf{R}}\big)^3\Big). \tag{27}$$

After differentiating contours and finding maximally connected components, we have $N^{\mathbf{R}} \ll N$. Further computation reduction can be via AP filtering and RP cluster mapping. Then we can significantly reduce the number of APs and RPs. In this way, INTRI can be integrated on existing onboard LBS systems and further support mobile targets.

5 Crowdsourced Parameter Estimation and Device Calibration

In this section, we study how to improve the accuracy of INTRI with online crowdsourced inputs and device calibration. In Section 5.1 we first propose a novel particle filter for parameter estimation of contour width. Then in Section 5.2, we describe an efficient online calibration in order to address device heterogeneity.

5.1 Crowdsourced Parameter Estimation

Offline training over z_0 for suitable contour width may not be scalable to online phase. Hence we consider as follows online parameter estimation to find suitable z_0. During the parameter learning phase, INTRI finds the target location based on some potential z_0's (say, from offline training samples or empirical studies). Then a crowdsourced input of her current location can be obtained through certain techniques like explicit user feedbacks or implicit sensor assistance (say, an iBeacon). Given input locations and corresponding RSSI vectors (say, in a batch), a particle filter is used for parameter estimation, i.e., to find the most suitable z_0's which produces location estimations consistent with those crowdsourced input ones. Gradually z_0 gets refined.

Note that we focus on the algorithmic design of contour width parameter estimation given any crowdsourcing schemes [6], [8]. The detailed input approach, related outlier detection and further error filtering over user inputs are orthogonal to our studies and can be referred to [6], [7], [36]. Their approaches can be easily applied as preprocessing of our system for accuracy improvement. In our experiment, we will also evaluate influence of crowdsourcing errors.

Algorithm 2. Contour-Based Indoor Trilateration

Input: $\{\mathbf{r}_n\}$ and $\{\mathbf{f}_n\}$: set of N RPs and the set of RSSI vector at each RP; $\mathbf{g}$: target RSSI vector; ζ: threshold in determining MCC; ρ: range of RPs to be selected.
Output: $\widehat{\mathbf{x}}$: estimated locations of the target.

/* Construction of Signal Contours. */
$\mathbf{C}^l \leftarrow \{\}$; /* Set of RPs with Contours. */
for $n \in \{1, \ldots, N\}$ **do**
  $D_n \leftarrow 0$;
  for $l \in \{1, \ldots, L\}$ **do**
    if $\bar{f}_n^l \geq g^l - z_0\sqrt{(\sigma^l)^2 + (\sigma_n^l)^2}$ *and* $\bar{f}_n^l \leq g^l + z_0\sqrt{(\sigma^l)^2 + (\sigma_n^l)^2}$ **then**
      $\mathbf{C}^l \leftarrow \mathbf{C}^l \cup \{n\}$; $D_n \leftarrow D_n + 10^{\frac{g^l - P_0^l}{10\gamma^l}}$;
    end
  end
  $D_n \leftarrow D_n / L$;
end
/* Selecting Dense Contours. */
$\mathbf{R} \leftarrow FindHighWeight(\mathbf{\Omega}, \rho, \{D_n\})$;
/* RP Filtering Using Connectivity. */
$\mathbf{R} \leftarrow FindMaxComponent(\mathbf{R}, \zeta)$;
/* Calculating Dist from Contours. */
for $n \in \{1, \ldots, N^{\mathbf{R}}\}$ **do**
  for $l \in \{1, \ldots, L\}$ **do**
    for $m \in \{1, \ldots, |\mathbf{C}^l|\}$ **do**
      $\delta_{nm}^l = \sqrt{(x_n - x_m^l)^2 + (y_n - y_m^l)^2}$;
    end
    $\Delta_n^l(\mathbf{g}, \mathbf{f}_n) = \min\{\delta_{nm}^l\}$;
  end
end
LP-based Localization Using Formulation (26);
$\widehat{\mathbf{x}} \leftarrow \sum_{n=1}^{N^{\mathbf{R}}} \omega_n \mathbf{r}_n$; /* Final Estimation. */

Specifically, uniform particle sampling within an interval around basic z_0 in Section 3.2 is first conducted. Then the locations using different z_0's are calculated by INTRI. We find the distance (or discrepancy) between estimated and the input locations from users in order to evaluate the consistency. Specifically, let ψ^p be the euclidean distance between estimated $\widehat{\mathbf{x}}_p$ and the input location $\mathbf{x}^i$, i.e.,

$$\psi^p = \|\widehat{\mathbf{x}}_p - \mathbf{x}^i\|, \quad p \in \{1, \ldots, P\}. \tag{28}$$

Then we calculate the weight θ_p of each particle based on the consistency between $\widehat{\mathbf{x}}_p$ and $\mathbf{x}^i$ as

$$\theta_p \leftarrow \frac{1}{\sqrt{2\pi}\sigma_w} \exp\left(-\frac{(\psi^p)^2}{2\sigma_w^2}\right), \tag{29}$$

where σ_w is the sensitivity of the weight. Hence θ_p represents the consistency between $\widehat{\mathbf{x}}_p$ and $\mathbf{x}^i$. In other words, those width parameters whose location estimations are closer to crowdsourced input ones get larger weights. Each weight will then be normalized, i.e., $\theta_p \leftarrow \frac{\theta_p}{\sum_{p=1}^{P} \theta_p}$.

Through resampling, the parameters with low consistency will be filtered due to the low weights [25]. The estimated parameter $\widehat{z}_0$ is given by the weighted average of the parameters generated from particles, i.e.,

$$\widehat{z}_0 \leftarrow \sum_{p=1}^{P} \theta_p z_0^p. \tag{30}$$

Given the estimated locations, INTRI finds the variance of particles, i.e., $\xi = Var(\psi^p)$. Then INTRI checks after each time of estimation that if the variance of particles satisfies $\xi \leq \eta$, where η indicates the predefined threshold, we can conclude that the width parameter estimation converges. Later INTRI is conducted based on the calibrated $\widehat{z}_0$ (i.e., $P = 1$ and $\widehat{z}_0$ becomes constant).

We briefly analyze computational complexity of particle filter here. Given P particles, the complexity of particle filter is $\mathcal{O}\Big(P\Big(NL + N^{\mathbf{R}}L|\mathbf{C}^l| + \big(N^{\mathbf{R}}\big)^3\Big)\Big)$ [25]. Further computation reduction can be conducted by AP filtering and RP cluster mapping [37] to reduce the number of APs and RPs. After calibration converges, $P = 1$, z_0 is fixed and online localization complexity of INTRI hence becomes small.

5.2 Online Calibration for Heterogeneous Devices

Due to difference in Wi-Fi network interfaces, for the same signal different smartphones may have different measurement values [19]. If such signal difference issue is not addressed, the contours (Section 3.2) cannot be found correctly. Leveraging such similarity, we present an efficient algorithm to adapt to different mobile devices as follows.

In this section we consider efficient and scalable online calibration in order to reduce offline manual efforts. To this end, we map the target signals $\mathbf{g}$ to the signal space in fingerprint database. We first calculate the correlation between the target RSSI vector and that of each RP. The RPs with similar signal vectors can be leveraged for online signal calibration. The vector comparison is based on the correlation between $\mathbf{g}$ and fingerprint $\mathbf{f}_n$, i.e.,

$$corr(\mathbf{g}, \mathbf{f}_n) = \frac{\sum_{l=1}^{L}\big(g^l - \bar{g}\big)\big(\bar{f}_n^l - \bar{f}_n\big)}{\sqrt{\sum_{l=1}^{L}\big(g^l - \bar{g}\big)^2}\sqrt{\sum_{l=1}^{L}\big(\bar{f}_n^l - \bar{f}_n\big)^2}}, \tag{31}$$

where $\bar{g} = \frac{1}{L}\sum_{l=1}^{L} g^l$ and $\bar{f}_n = \frac{1}{L}\sum_{l=1}^{L} \bar{f}_n^l$. The above correlation compares relative signal trend of different APs rather than the absolute RSSI values. Based on Equation (31), we can find the RPs with similar signal trend for calibration and reduce the effect of device dependency.

To mitigate the effect of random noise, we find the top several RPs with $corr(\mathbf{g}, \mathbf{f}_n) > \eta$ ($\eta = 0.95$ in our experiment) for linear calibration. For each target RSSI g^l from AP l, we find the corresponding f_n^l at RPs. Given pairs of $[g^l, f_n^l]$, we conduct the linear regression and obtain the corresponding a and b for target RSSI g^l, i.e.,

$$\widetilde{g}^l = ag^l + b. \tag{32}$$

Note that our online calibration approach is not restricted to linear model, and is general enough to apply to any other signal mapping model (e.g., [18]). Based on Equation (32) we can conduct INTRI with calibrated $\widetilde{g}^l$. Given $\mathcal{O}(L)$ APs and $\mathcal{O}(N)$ RPs, the correlation comparison takes $\mathcal{O}(NL)$. Let λ ($\lambda \ll N$) be the number of RPs whose correlation $corr(\mathbf{g}, \mathbf{f}_n)$ is greater than η, and linear regression takes $\mathcal{O}(L^2\lambda^2)$. Therefore, the online computational complexity of 2-D linear regression is $\mathcal{O}(NL + L^2\lambda^2)$ [35].

6 Experimental Evaluations

We have conducted extensive experimental trials of INTRI in the Hong Kong International Airport (HKIA), our university campus (HKUST) and the Hong Kong Cyberport (HKCP). We first present our settings in Section 6.1. As the measured AP signals are different in the three sites, we discuss the comparative studies over these differences in Section 6.2. Then we illustrate the experimental results with Wi-Fi and Bluetooth iBeacon (Section 6.3), followed by validation of crowdsourced user inputs (Section 6.4).

6.1 Experimental Settings

We use in our experimental studies the same state-of-the-art algorithms and comparison metrics as follows. Besides, we compare the device calibration scheme in INTRI with two recent methods, signal strength difference (SSD) [19] and signal ratio (SR) [20]. SSD utilizes the differences between pairs of AP signal values as patterns. Similarly, SR calculates the ratio between pairs of AP signals as Wi-Fi fingerprints. Both methods aim at compensating the signal difference among heterogeneous devices.

Algorithm 3. User-Assistance for Contour-Based Localization

Input: P: number of particles for consistency check.
$\boldsymbol{g}$ and $\{\boldsymbol{f}_n\}$: RSSIs at target and RPs.
Output: Ω: set of parameters for INTRI.
$\Omega \leftarrow \{\}$; /* `Initialization` */
if *NoParticles* **then**
  for $p \leftarrow 1$ to P **do**
    $z_0^p \leftarrow$ RandSam($[z_0^{min}, z_0^{max}]$); Add z_0^p into Ω;
  end
end
$\{\widehat{\mathbf{x}}_p\} \leftarrow$ Localization Using Formulation (26);
for $p \leftarrow 1$ to P **do**
  $\psi^p \leftarrow \|\widehat{\mathbf{x}}^p - \mathbf{x}^i\|_2$;
  $\theta_p \leftarrow \exp\Big(-(\psi^p)^2/\big(2\sigma_w^2\big)\Big)/\big(\sqrt{2\pi}\sigma_w\big)$;
  /* `Particle Weight Recalculation` */
end
/* `Resampling of Width Parameter` */
for $p \leftarrow 1$ to P **do**
  $\{z_0^p, \theta_p\} \leftarrow \text{Resample}(\Omega)$;
end
Normalize($\{\theta_p\}$); /* `Normalization` */
$\widehat{z}_0 \leftarrow \sum_{p=1}^{P} \theta_p z_0^p$; /* `Parameter Estimate` */

We compare INTRI with several state-of-the-art schemes: 1) *EZPerfect* (**EZPerf**) [13], [14]: a model-based Wi-Fi localization scheme which considers deriving signal propagation model from fingerprint data. Given Wi-Fi fingerprints, EZPerfect first finds the matching relationship between signals and distances from APs [13], and then locates targets with a genetic algorithm solving trilateration problem [14]. 2) *KL-divergence* (**KL-div**) [4], [11]: which utilizes the Kullback-Leibler (KL) divergence distance [4] between the signal distribution at an RP and target during comparison. Then the top k RPs with the minimum KL-divergence are utilized for final estimation. 3) *Weighted k-Nearest Neighbors*

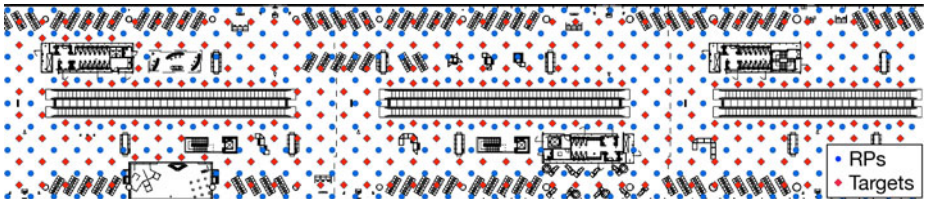


Fig. 6. Indoor map of a boarding gate at the HKIA. The survey grid size is 5 m (survey conducted on July 8, 2014).

(**WKNN** [1], [12]): It computes the cosine similarity [12] between the fingerprints and the target RSSI vector. Then it finds the weighted average of k-Nearest Neighbors [1] of highest cosine similarity to estimate the target location. 4) *Probabilistic algorithm* (**Horus** [9]): It first calculates the probability distribution of the RSSI values at each RP. Given a target RSSI vector, Horus computes the overall probability of the vector at each RP and finds the top several ones with the maximum likelihood as the target location. 5) iBeacon fingerprinting (**MAP** [38]): For iBeacon fingerprinting localization, we also compare INTRI with *MAP* [38] in addition to above algorithms. MAP leverages the maximum a-posterior probability mapping over Bluetooth RSSI fingerprints.

Fig. 6 shows the corresponding survey floor plan of RPs and targets in HKIA. In the $10{,}000\ \mathrm{m}^2$ site, we collect 340 RPs and 1,100 targets. We utilize HTC One X as the fingerprint collector and Lenovo A680 for target measurement. The locations of RPs are predefined on the indoor map. To balance between localization accuracy and survey cost, we use 5 m grid density in fingerprinting. At each RP, signal data is sampled from four different directions (north, west, south and east). For each direction, 15 samples of RSSI vectors are collected. The ground truth of the target locations is also predetermined in grid form (in the testing, the surveyors find the RP or target locations from the nearest pillars, floor tiles and other noticeable indoor landmarks). Note that the data sampling is conducted with people around, and temporal fluctuation exists within the fingerprints and target signals. The time interval between samples in Wi-Fi scanning is 1 second.

Wi-Fi APs are officially pre-deployed. Their number, locations and transmission power are already settled before site survey. During preprocessing, we filter out the mobile APs tethered by smartphones, and combine the signals of virtual APs (VAPs). Overall 360 APs are detected (each RP detects 47 APs on average). Part of these APs may be installed outside the survey site since their coverage in our site is relatively small and signals are globally weak. The target samples are collected one month later than RP collection. For schemes like KL-divergence and Horus which are device dependent, we utilize our scheme to calibrate the signals.

In the HKUST campus and the HKCP shopping mall, fingerprint collection, target sampling and data preprocessing are the same as those in the airport. Fig. 7 shows the RPs and targets on campus ($100 \times 50m^2$). In campus corridor environment, we collect 250 RPs and 475 targets. Fig. 8 shows the RPs and targets in the shopping mall ($150 \times 100m^2$). In the HKCP mall, we collect 680 RPs and 680 targets. In both the HKUST and HKCP, the blue dots represent RPs and the red diamonds are targets. Similar to the HKIA, we are given the officially deployed APs and we cannot manually change their settings (installation locations and TX power). In the site survey of campus corridor, overall 320 APs are detected (each RP measures 24 APs on average). In the site survey at shopping mall, overall 190 APs are detected (average 28 APs at each RP).

We have also conducted extensive studies using iBeacon (with TI CC2541) with INTRI on the HKUST campus hall (as in Fig. 9a). We have deployed 7 iBeacons in the campus hall. 135 targets and 506 reference points are collected on Oct 27th, 2015 with Apple iPhone 6. As shown in Fig. 9b, we have deployed iBeacons on walls and pillars of the campus hall. Similar fingerprinting like Wi-Fi survey in Fig. 7 is also conducted there. For each iBeacon, we set its transmission power at default -12 dBm and the beaconing interval as 500 ms. Note that our system and device calibration can be easily extended to scenarios when Wi-Fi and Bluetooth are deployed together, as our contour approach does not need to differentiate Wi-Fi from Bluetooth.

To evaluate the improvement by user inputs, we have further collected new target RSSIs and user input data on HKUST and HKIA. A user inputs current $<$location,

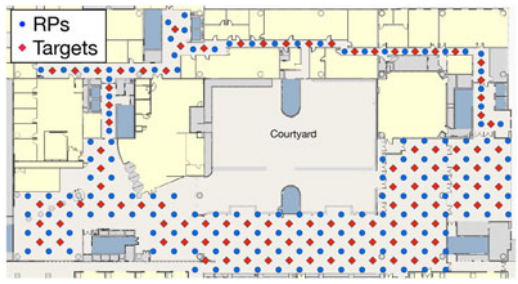


Fig. 7. Hall map of HKUST campus (survey conducted on Nov. 28, 2014).

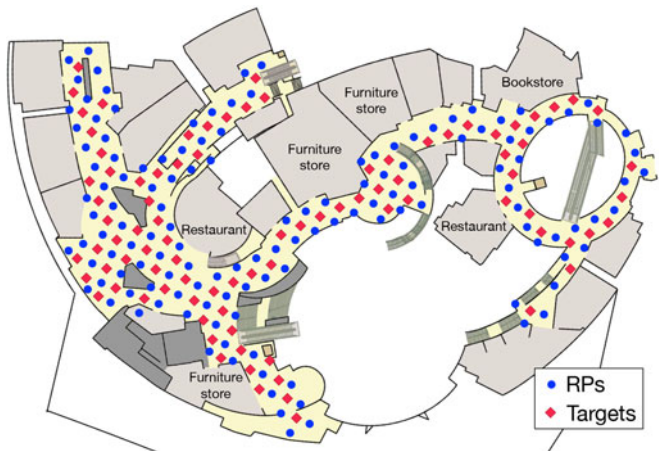


Fig. 8. Indoor map of HKCP mall (survey on Sept. 10, 2014).

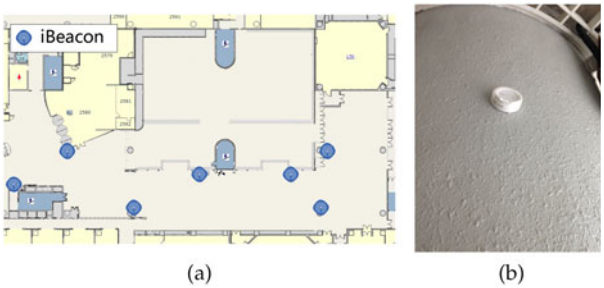


Fig. 9. (a) Locations of iBeacons on HKUST campus environment. (b) An iBeacon on a pillar (campus).

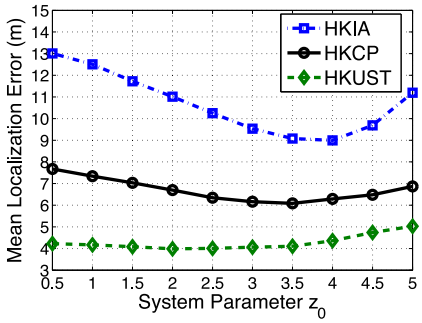


Fig. 10. Mean localization errors versus parameter z_0 in three sites.

RSSI > when using INTRI at different locations. 140 particles are implemented for width parameter calibration. z_0 is initialized between $[z_0^{min}, z_0^{max}] = [1.5, 4.5]$. In HKUST, 80 users are introduced at each arrival, while in HKIA 110 users are introduced. After each time of particle filter estimation, we evaluate their mean localization error. Meanwhile, we also present the batch offline updating given manually-collected test data, which may serve as the ground-truth comparison. We further evaluate the effect of user-input variance (with the location error following $\mathcal{N}(0, \sigma_{in}^2)$ between the ground truth and the input one), and at the baseline $\sigma_{in} = 3$ m is introduced in crowdsourced input locations. We also show the mean localization error of INTRI without crowdsourced assistance as the benchmark (with fixed $z_0 = 2$).

6.2 Comparative Studies of Different Sites

We first summarize the signals at the experimental sites as follows. On average, each target can detect 16 APs on HKUST and 22 APs in HKCP. In HKIA, each target can detect 16 APs on average. Though targets in HKIA and HKUST have similar detected AP number, the survey site in HKUST is smaller and hence it has denser AP deployment. Moreover, the signal coverage of APs in our campus corridor and HKCP is constrained by the wall partitions, which helps differentiate the RPs. Therefore, we expect a better localization performance on HKUST and HKCP than in HKIA. Based on these detected APs, we evaluate the effect of different received AP numbers. In our site survey, we can also observe a smaller signal noise in HKUST and HKCP than that in HKIA. It is because the airport boarding area is large open space with many airline passengers, which leads to higher signal uncertainty. Based on the difference in signal noise, we adjust the parameter z_0 in

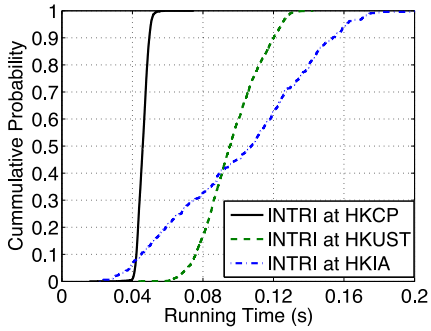


Fig. 11. Cumulative probability of INTRI running time on different sites.

Equation (10) for online localization. Further illustration of the signals in the experimental sites can be referred to [5].

Fig. 10 shows the mean localization errors of INTRI in three sites given different z_0 in setting z_0. In general, the error first decreases and then increases. This is because the localization error depends on two factors: signal noise and RP fingerprint differentiation. When z_0 is small, the tight contours cannot accommodate the large measurement uncertainty in target. Thus, the error is high. As z_0 increases, the contours can bound the target, and hence the error decreases. As z_0 further increases, the error increases because, as contours become wide, the differentiation between RPs becomes weak. Then more distant RPs are included in localization. The result shows that without sufficient RP fingerprint differentiation, wide contours would not help. Compared with HKUST and HKCP, z_0 is slightly larger in HKIA due to higher signal variance in airport. Thus, in our experiment, initially $z_0 = 4$ in HKIA, $z_0 = 3.5$ in HKCP, and $z_0 = 2.5$ on our HKUST campus. Further fine-grained z_0 is obtained through particle filter later (Section 6.4) to achieve online environmental adaptation.

Fig. 11 shows running time CDF of INTRI on targets at different sites. We test INTRI on a server PC with i7 4790 (3.6 GHz) CPU. It shows that at the three sites, our INTRI shows different computation time due to the difference in RP numbers and detected AP number at target and RP. The running time of INTRI at HKCP is much smaller than that at HKIA and HKUST. It is mainly because many of detected APs in HKCP are installed in the shop stores and have limited coverage (small contours) due to wall partitions, compared with those in HKIA and HKUST. A target can be quickly mapped to smaller areas and therefore the overall computation becomes much smaller. For each INTRI component, the computation time (percentage in the total time) on average is as follows: finding and filtering the contours (Section 3) takes 20.9 ms (18.19 percent) per query; the linear programming (Section 4) takes 94.1 ms (71.81 percent) per location query. The major online computation comes from the contour finding and the linear programming. The particle filter (Algorithm 3) takes 13.32 s for each time of crowdsourced learning (80 users). Note that the particle filter can be conducted at the backend server and therefore its influence over the user experience is minimum.

6.3 Experimental Evaluation with Wi-Fi & iBeacon

Fig. 12 shows the linear signal model of four different target estimations in HKIA. Target data (x-axis) are collected using Lenovo A680 while RP signals (y-axis) are from HTC One X.

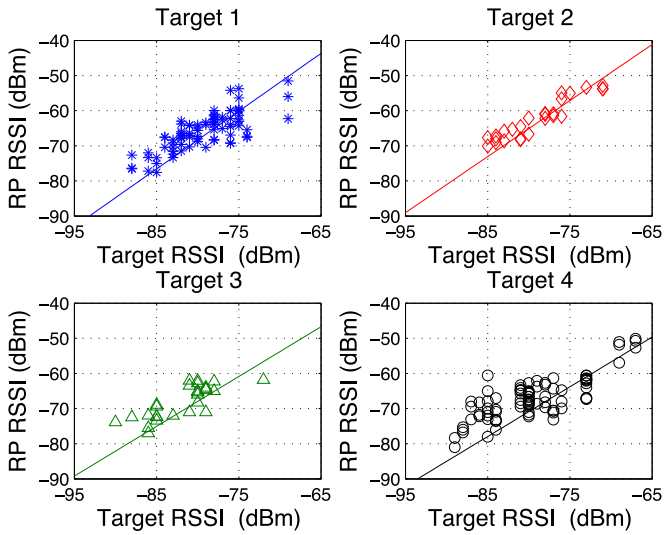


Fig. 12. Calibration of target RSSI from RP signals (HKIA).

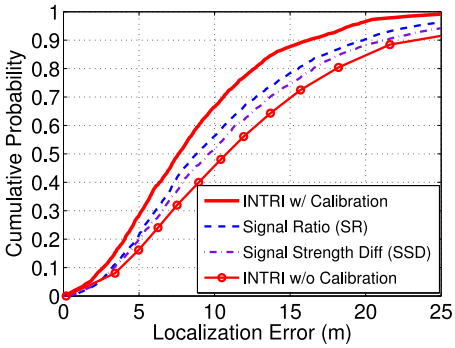


Fig. 13. Performance of different device calibration methods in HKIA.

We obtain the calibrated signal strength, $\tilde{g}^l$, for each given target. We can see that the linear calibration scales up the Lenovo A680 measurements (target).

Fig. 13 compares the cumulative errors of different device calibration schemes in the airport. We use uncalibrated signals in INTRI as the baseline case. The calibrated INTRI improves from the uncalibrated scheme, and also outperforms SSD and SR. It is mainly because our proposed scheme jointly considers the relative trend and the RSSI adjustment model when calibrating devices. Unlike SSD and SR, our correlation scheme mitigates the errors in signal values when INTRI calibrates target RSSI using fingerprints, which is more robust under large signal noise in the airport.

Fig. 14 shows the localization error with and without differentiating the contours using HKIA data. We consider two scenarios using 5 m (default) and 10 m survey grid size. Without using contour weights, we count the number of signal contours and implement it into INTRI as the baseline case. As shown in Fig. 14, simple contour counting cannot discriminate the dispersed nearest neighbors (RPs) with similar number of contours. In contrast, contour weight discriminates the fingerprints by penalizing RPs only with many weak signals. Then we mitigate the influence from the noise in the measurement. Thus, we have shown using contour weights achieves better performance than the unweighted scheme, especially under sparser survey grid size.

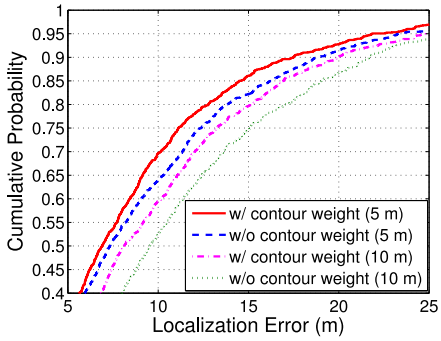


Fig. 14. Performance of INTRI with and without contour weights (HKIA).

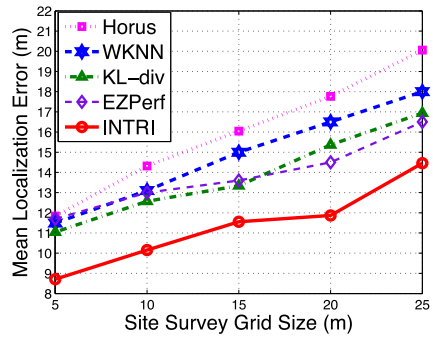


Fig. 15. Mean localization errors versus site survey grid size (HKIA).

Fig. 15 shows the mean localization errors against the survey grid size. As the minimum grid size is five meters, columns or rows of RPs are removed to form the grid sizes with multiples of five. Clearly all five algorithms degrade as grid size increases. We can observe EZPerfect achieves slightly higher accuracy than WKNN, KL-divergence and Horus under larger grid size. It is because for EZPerfect additional distances from multiple APs constrain the target location and prevent large error. Compared with the algorithms above, INTRI has much higher localization accuracy at different grid sizes. It is because INTRI considers signal uncertainty in contours and constrains the target estimation through joint optimization. Without relying on accurate distance measurement, INTRI can still constrain the target estimation by signal contours under large grid size.

Fig. 16a compares the cumulative errors of five algorithms in HKIA. Due to large measurement noise in the airport, WKNN's accuracy is weakened by the dispersed nearest neighbors. Horus assumes a certain distribution of signal level at each RP and therefore cannot represent real signal distribution under limited sampling. KL-divergence also requires large data sampling and dense fingerprints in signal distribution comparison. Therefore, it cannot adapt to the noisy airport environment. The large signal noise also degrades the distance accuracy of traditional trilateration in EZPerfect. In contrast to above methods, INTRI jointly considers the signal noise and distances to contours, and therefore reduces misestimation of the target.

Figs. 16b and 16c show the cumulative errors in HKUST and HKCP, respectively. Compared with the HKIA, the fingerprints and target signals in HKUST and HKCP carry less signal measurement noise. Thus, we observe the increase of localization accuracy in all the algorithms at these two sites

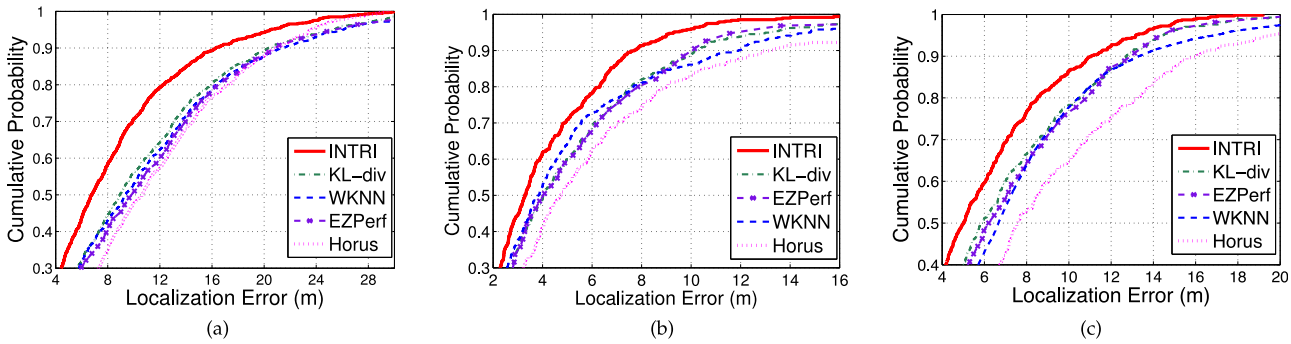


Fig. 16. Cumulative probability of localization errors in (a) HKIA boarding gate, (b) HKUST campus, and (c) HKCP mall.

compared with that in the airport. EZPerfect becomes slightly better than WKNN as the distance measurement becomes less noisy. Similar to Fig. 16a, INTRI achieves higher accuracy than other algorithms since it considers the signal variation in constructing contours and utilizes them to reduce the dispersed nearest neighbors. INTRI is general enough to work under different environments with markedly higher accuracy. We also study the performance of INTRI in the HKUST and HKCP extensively. As the conclusions are qualitatively the same, for brevity we will not repeat them here.

Fig. 17 shows the cumulative probability of localization error with iBeacon in our campus hall. Our INTRI outperforms other schemes as it constrains the location with contours of Bluetooth signals, which jointly considers signal strength and variance. Compared with Wi-Fi, our iBeacon experiment on INTRI achieves higher accuracy, as the Bluetooth signals from iBeacon have smaller coverage, faster signal degrade and better differentiation than Wi-Fi on campus. In practice, fusing Wi-Fi and iBeacon may be an approach to resolve the accuracy and cost (infrastructure and management) issue.

Fig. 18 shows mean error of iBeacon-based positioning versus the survey grid size, which represents the fingerprint signal density. Clearly, as the density decreases, localization error decreases. With contour constraints, our scheme achieves better accuracy under all survey density.

6.4 Experimental Evaluation on Crowdsourced Inputs

Fig. 19 shows the mean localization error (upper figure) and estimated width parameter z_0 (lower figure) versus index of user feedback arrivals. Note that the mean localization error in fact has deteriorated from 4 m (see Fig. 10) to above 5 m at the initialization without crowdsourcing inputs. At each batch of user arrival, we introduce 80 user feedbacks. We can observe that as more users are involved, the mean localization error of all incoming users decreases. When a certain number of feedbacks are provided (given two batches of crowdsourced inputs, i.e., 160 users), the accuracy of INTRI begins to converge due to sufficient calibration of the system parameters. We can observe that INTRI with user assistance achieves lower errors than without user feedbacks. It is mainly because the fixed z_0 which is trained offline may not sufficiently reflect the online signals. We can also see that the performance of the proposed crowdsourced learning closely matches with using the offline manually-collected test data (i.e., batch offline updating).

Fig. 20a shows the mean localization error versus the location input error (standard deviation σ_{in} in the user-input locations). We can observe that as σ_{in} increases, the localization error also increases due to misestimation of width parameters. However, the trend of increase is overall small even under large positioning errors. Our scheme is robust towards the input noise, as the particle filter refines and finds those width parameters which provides consistent matching of locations given long-term user inputs. Fig. 20b shows the mean localization error versus the number of particles. As more particles are used, INTRI gets better calibrated and hence the localization error decreases. Trade-off between location accuracy and computation efficiency exists. For practical deployment, we choose a certain number of particles (say, 140) as baseline in our experimental deployment.

We have also conducted extensive experimental studies in HKIA. Fig. 21 shows the localization accuracy and the estimated width parameter versus the user arrivals in the airport. We can also see the localization error decreases as more user feedbacks are used for parameter adaptation and location accuracy improvement. Given certain number of user feedbacks (say, 4 batches or 320 user inputs), the

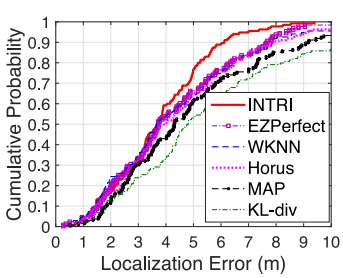


Fig. 17. CDF of localization errors with iBeacon (HKUST).

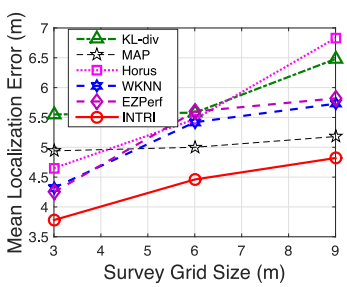


Fig. 18. Mean localization error versus survey grid size with iBeacon (HKUST).

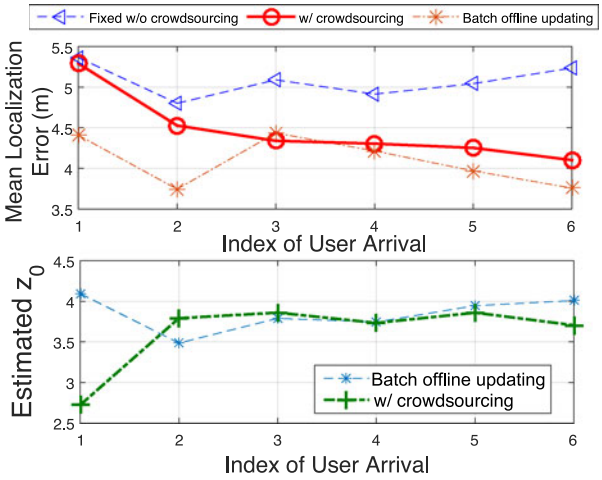


Fig. 19. Mean localization error (m) and width parameter z_0 versus index of user arrivals (HKUST).

localization errors already converge. As other results are qualitatively similar to those in HKUST campus, for brevity of description we do not repeat them here.

7 Conclusion

Traditional trilateration has achieved much success for outdoor localization. However, it does not work well indoors due to non-line-of-sight measurement and signal fading. Fingerprinting is a promising approach for indoor localization, but its performance is vulnerable to signal noise. We propose in this work INTRI, a highly accurate algorithm which combines the advantages of both trilateration and fingerprinting to achieve much better accuracy.

Based on the spirit of trilateration, for each measured AP signal level, INTRI forms the corresponding contour, which is the set of RPs with the same signal level subject to its statistical fluctuation. The target is hence where the contour is. To estimate target's location, INTRI formulates a linear programming to minimize the distance between the location and these contours (i.e., following the spirit of trilateration). To achieve higher accuracy, INTRI addresses device heterogeneity with an efficient and scalable algorithm based on the correlation in RSSI for online signal calibration. In order to

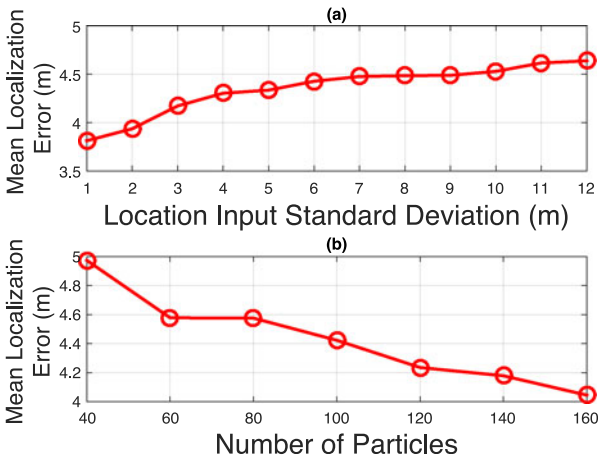


Fig. 20. Mean localization error (m) versus (a) location input deviation σ_{in}; (b) number of particles (HKUST).

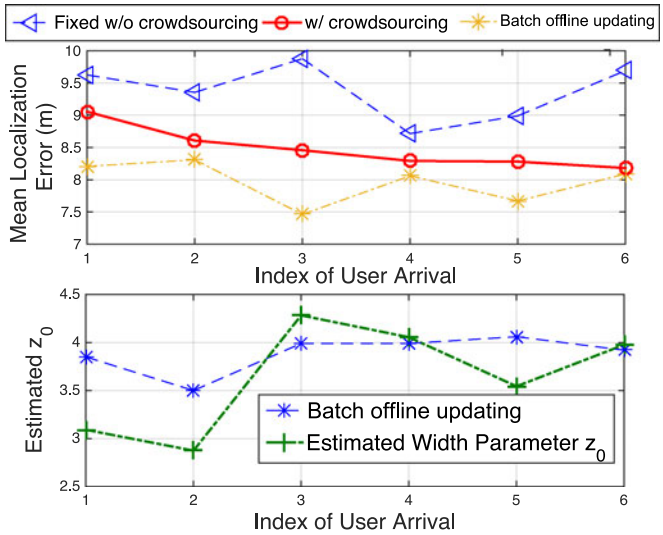


Fig. 21. Mean localization error (m) and width parameter z_0 versus number of user arrivals (HKIA).

adapt towards deployment environment we further implement crowdsourced inputs of $<$locations, signals$>$ to estimate the contour width parameter. A novel particle filter is applied to achieve this. We have conducted extensive simulation and experimental Wi-Fi and iBeacon trials on INTRI in the Hong Kong International Airport, Hong Kong Cyberport mall and HKUST campus. Compared with other approaches, INTRI achieves significantly higher accuracy and robustness under signal noise (often by more than 20 percent), and adapts to environments with crowdsourced assistance.

Acknowledgments

This work was supported, in part, by the Hong Kong Research Grant Council (RGC) General Research Fund (610713).

References

[1] D. Han, S. Jung, M. Lee, and G. Yoon, "Building a practical Wi-Fi-based indoor navigation system," *IEEE Pervasive Comput.*, vol. 13, no. 2, pp. 72–79, Apr. 2014.

[2] L. Li, G. Shen, C. Zhao, T. Moscibroda, J.-H. Lin, and F. Zhao, "Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service," in *Proc. ACM 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 459–470.

[3] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, 2000, vol. 2, pp. 775–784.

[4] P. Mirowski, "Probability kernel regression for WiFi localisation," *J. LBS*, vol. 6, no. 2, pp. 81–100, 2012.

[5] S. He, T. Hu, and S.-H. G. Chan, "Contour-based trilateration for indoor fingerprinting localization," in *Proc. ACM Conf. Embedded Netw. Sens. Syst.*, 2015, pp. 225–238.

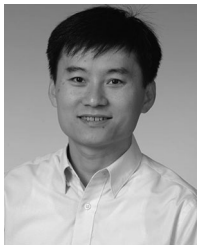
[6] J.-G. Park, et al., "Growing an organic indoor location system," in *Proc. ACM 8th Int. Conf. Mobile Syst. Appl. Serv.*, Jun. 2010, pp. 271–284.

[7] Y.-C. Chen, J.-R. Chiang, H.-h. Chu, P. Huang, and A. W. Tsui, "Sensor-assisted Wi-Fi indoor location system for adapting to environmental dynamics," in *Proc. ACM Int. Symp. Model. Anal. Simulation Wireless Mobile Syst.*, 2005, pp. 118–125.

[8] M. Lee and D. Han, "QRLoc: User-involved calibration using quick response codes for Wi-Fi based indoor localization," in *Proc. IEEE 7th Int. Conf. Comput. Convergence Technol.*, Dec 2012, pp. 1460–1465.

[9] M. Youssef and A. Agrawala, "The Horus location determination system," *Wireless Netw.*, vol. 14, no. 3, pp. 357–374, 2008.
[10] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 689–705, Jun. 2007.
[11] P. Mirowski, D. Milioris, P. Whiting, and T. Kam Ho, "Probabilistic radio-frequency fingerprinting and localization on the run," *Bell Labs Tech. J.*, vol. 18, no. 4, pp. 111–133, 2014.
[12] S. Han, C. Zhao, W. Meng, and C. Li, "Cosine similarity based fingerprinting algorithm in WLAN indoor positioning against device diversity," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 4313–4317.
[13] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. ACM 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 173–184.
[14] R. Nandakumar, K. K. Chintalapudi, and V. N. Padmanabhan, "Centaur: Locating devices in an office environment," in *Proc. ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 281–292.
[15] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-Markie: Indoor pathway mapping made easy," in *Proc. 10th USENIX Conf. Netw. Syst. Des. Implementation*, 2013, pp. 85–98.
[16] Y. Jiang, et al., "Hallway based automatic indoor floorplan construction using room fingerprints," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 315–324.
[17] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. ACM 10th Int. Conf. Mobile Syst. Appl. Serv.*, 2012, pp. 197–210.
[18] J.-G. Park, D. Curtis, S. Teller, and J. Ledlie, "Implications of device diversity for organic localization," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 3182–3190.
[19] A. Mahtab Hossain, Y. Jin, W.-S. Soh, and H. N. Van, "SSD: A robust RF location fingerprint addressing mobile devices' heterogeneity," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, pp. 65–77, Jan. 2013.
[20] M. B. Kjaergaard, "Indoor location fingerprinting with heterogeneous clients," *Pervasive Mobile Comput.*, vol. 7, pp. 31–43, 2011.
[21] S.-H. Fang, C.-H. Wang, S.-M. Chiou, and P. Lin, "Calibration-free approaches for robust Wi-Fi positioning against device diversity: A performance comparison," in *Proc. IEEE VTC Spring*, 2012, pp. 1–5.
[22] S. Yang, P. Dessai, M. Verma, and M. Gerla, "FreeLoc: Calibration-free crowdsourced indoor localization," in *Proc. IEEE INFOCOM*, 2013, pp. 2481–2489.
[23] D. H. Kim, K. Han, and D. Estrin, "Employing user feedback for semantic location services," in *Proc. ACM 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 217–226.
[24] A. K. M. Mahtab Hossain, H. Nguyen Van, and W.-S. Soh, "Utilization of user feedback in indoor positioning system," *Pervasive Mob. Comput.*, vol. 6, no. 4, pp. 467–481, Aug. 2010.
[25] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
[26] E. Bhasker, S. Brown, and W. G. Griswold, "Employing user feedback for fast, accurate, low-maintenance geolocationing," in *Proc. IEEE Annu. Conf. Pervasive Comput. Commun.*, Mar. 2004, pp. 111–120.
[27] L. Koping, M. Grzegorzek, F. Deinzer, S. Bobek, M. Slazynski, and G. Nalepa, "Improving indoor localization by user feedback," in *Proc. IEEE 18th Int. Conf. Inf. Fusion*, July 2015, pp. 1053–1060.
[28] A. Mtibaa, K. A. Harras, and M. Abdellatif, "Exploiting social information for dynamic tuning in cluster based WiFi localization," in *Proc. IEEE Int. Conf. Wireless Mobile Comput. Netw. Commun.*, Oct. 2015, pp. 868–875.
[29] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, "Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 147–158.
[30] M. Youssef and A. Agrawala, "Handling samples correlation in the Horus system," in *Proc. IEEE INFOCOM*, 2004, pp. 1023–1031.
[31] B. Ferris, D. Fox, and N. D. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 2480–2485.
[32] S. S. Skiena, *The Algorithm Design Manual*. Berlin, Germany: Springer, 2008.
[33] JOptimizer, JOptmizer—java convex optimizer, 2016. [Online]. Available: http://www.joptimizer.com/index.html
[34] CVXOpt, Python software for convex optimization, 2016. [Online]. Available: http://cvxopt.org/
[35] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[36] P. Bolliger, "Redpin-adaptive, zero-configuration indoor localization through user collaboration," in *Proc. ACM MELT*, 2008, pp. 55–60.
[37] S. He and S. H. G. Chan, "Tilejunction: Mitigating signal noise for fingerprint-based indoor localization," *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1554–1568, Jun. 2016.
[38] R. Faragher and R. Harle, "Location fingerprinting with Bluetooth low energy beacons," *IEEE J. Selected Areas Commun.*, vol. 33, no. 11, pp. 2418–2428, Nov. 2015.

Suining He received the BEng degree (highest honor) from the Huazhong University of Science and Technology, Wuhan, Hubei, China, in 2012. He is currently working towards the PhD degree in the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. His research interest includes indoor localization and mobile computing.



S.-H. Gary Chan (S'89-M'98-SM'03) received the BSE (highest honor) degree in electrical engineering from Princeton University, Princeton, New Jersey, in 1993, with certificates in applied and computational mathematics, engineering physics, and engineering and management systems. He received the MSE and PhD degrees in electrical engineering from Stanford University, Stanford, California, in 1994 and 1999, respectively, with a minor in business administration. He is currently a professor in the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (HKUST), Hong Kong. He is also a chair of the Task Force on Entrepreneurship Education, HKUST. His research interests include multimedia networking, wireless networks, mobile computing, and IT entrepreneurship. He has been an associate editor of *the IEEE Transactions on Multimedia* (2006-2011), and a vice-chair of Peer-to-Peer Networking and Communications Technical Sub-Committee of IEEE Comsoc Emerging Technologies Committee (2006-2013). He is and has been a guest editor of the *ACM Transactions on Multimedia Computing*, the *Communications and Applications* (2016), the *IEEE Transactions on Multimedia* (2011), the *IEEE Signal Processing Magazine* (2011), the *IEEE Communication Magazine* (2007), and the *Springer Multimedia Tools and Applications* (2007). He was the TPC chair of IEEE Consumer Communications and Networking Conference (IEEE CCNC) 2010, Multimedia symposium of IEEE Globecom (2007 and 2006), IEEE ICC (2007 and 2005), and Workshop on Advances in Peer-to-Peer Multimedia Streaming in ACM Multimedia Conference (2005). He has co-founded several startups with his team based on his research. His projects received industrial innovation awards in the Hong Kong, Pan Pearl River Delta, and Asia-Pacific regions due to their commercial impacts (2012-2015). He received the Google Mobile 2014 Award (2010 and 2011) and Silver Award of Boeing Research and Technology (2009). He has been a visiting professor and researcher with Microsoft Research (2000-2011), Princeton University (2009), Stanford University (2008-2009), and the University of California at Davis (1998-1999). He was undergraduate programs coordinator of the Department of Computer Science and Engineering (2013-2015), director of Sino Software Research Institute (2012-2015), co-director of Risk Management and Business Intelligence program (2011-2013), and director of Computer Engineering Program (2006-2008), HKUST. He was a William and Leila fellow with Stanford University (1993-1994), and received the Charles Ira Young Memorial Tablet and Medal, and the POEM Newport Award of Excellence at Princeton (1993). He is a member of honor societies Tau Beta Pi, Sigma Xi, and Phi Beta Kappa. He is a senior member of the IEEE.